

## OAI-PMH Extensions & other advanced functionality

In addition to the OAI Protocol's use as a basic mechanism for sharing metadata between data providers and service providers, a number of projects and researchers have been experimenting with extensions and novel uses for the protocol which go well beyond its original intentions. Samplings of those projects are discussed below. In addition this chapter will discuss possible future directions for the OAI-PMH.

### ***Extensible Repository Resource Locators (ERRoLs)***

Extensible Repository Resource Locators (ERRoLs)<sup>1</sup> are persistent or “Cool URLs”<sup>2</sup> for metadata, content, and services related to registered OAI-PMH repositories. The concept was developed by Jeff Young at OCLC Research. OCLC Research currently has an ERRoL resolution service with a base URL of <http://errol.oclc.org/>. However, it would be relatively easy to implement other ERRoL services as well. The source code for OCLC's ERRoL service is currently available as Open Source from <http://www.oclc.org/research/software/oai/errol.htm>. The genesis for ERRoLs came from PURL-based Object Identifiers (POIs)<sup>3</sup>, “Using the OAI-PMH ... Differently”<sup>4</sup>, and the OAI Registry at UIUC<sup>5</sup>.

All ERRoLs depend on a registration authority, such as the OAI Registry at UIUC. The registration authority maintains a mapping between the OAI repository identifiers and the OAI base URLs of the repositories. These mappings need to be accessible to the ERRoL resolver. This is done by making the registry into an OAI data provider which then allows the ERRoL resolver to gather the required mappings either as needed or in a batch. If the OAI repository follows the standard for OAI-Identifiers,<sup>6</sup> then that identifier is used for the repository. However, even for

repositories that do not conform to the OAI-Identifiers standard, identifiers can still be manually assigned to those repositories.

ERRoLs can be created for both OAI items and OAI repositories. All ERRoLs begin with a base URL (for the OCLC resolver the base URL is <http://errol.oclc.org/>) followed by an identifier which is either the repository identifier or an OAI item identifier. For example:

"http://errol.oclc.org/" + <repositoryIdentifier>

"http://errol.oclc.org/" + <oai-identifier>

"http://errol.oclc.org/" + <repositoryIdentifier> + "/" + <identifier>

The first example is a Repository ERRoL. The middle example is an Item ERRoL. The last example is an Item ERRoL for an OAI repository that does not use standard OAI-Identifiers, but does have a manually assigned repository identifier maintained by the registration authority.

The entire above are considered to be “Core ERRoLs.” A Core Repository ERRoL acts as a surrogate for the actual base URL of the repository in that you can append any normal OAI-PMH query parameter to the ERRoL, and the entire query will be redirected to the actual repository base URL. A Core Item ERRoL will be redirected to the repository base URL as a **GetRecord** request for the oai\_dc format for the item identified by the ERRoL. Core ERRoLs are useful primarily as persistent or “Cool URLs.” They can continue to be used even if the original base URL of a repository changes, assuming that the repository has not changed its repository identifier and the record for the repository has been updated in the central registration authority.

However, Core ERRoLs may be followed by any of a number of extensions to create an “Extended ERRoL.” The extensions are what give ERRoLs their real power. They can be used to resolve to related content, metadata, and services for the item or repository. Examples of extensions that have been implemented by OCLC’s ERRoL resolver include:

“.html” to create an HTML view of the record or repository,

“.ListMetadataFormats” to list the available metadata formats for the item,  
“.resource” to redirect to the first resolvable dc:identifier contained in the record,  
“.rss” to return an RSS feed for the repository,  
“.v10,” “.v11”, and “.v20” will act as surrogate base URLs for a repository, but responses from the repository will be converted into the appropriate version of the protocol.  
“.ListERRoLs” to list all available Extended ERRoLs that are available for the item or repository, and  
“.<metadataPrefix>” to return the raw XML response to the **GetRecord** for the given metadata format without the OAI wrapper elements.

This last extension of “.<metadataPrefix>” is particularly interesting because it allows systems to be developed for clients that may not understand the OAI Protocol at all. The example given at the ERRoL web site<sup>7</sup> posits a repository of XML Schemas where the a metadata prefix of “xsd” returns the actual XML Schema and the metadata prefix of “xhtml” returns the human-readable, XHTML description of that schema. This allows people to use URLs such as:

<http://errol.oclc.org/oclc:xmlregistry.oclc.org:errol/customERRoLSchema.xsd> or  
<http://errol.oclc.org/oclc:xmlregistry.oclc.org:errol/customERRoLSchema.xhtml>

without needing to even be aware that the objects they are requesting are being pulled directly from an OAI repository. It is easy to imagine other types of ERRoL extensions that might provide various services for OAI records, such as an extension like “.nameAuth” for generating possible name authority records for any dc:creator or dc:contributor elements contained in the record.

A number of OAI-related projects have already begun using ERRoLs or the concepts behind ERRoLs. Several novel repositories at OCLC use ERRoLs to provide an HTML user front end, such as the OCLC SchemaTrans Crosswalk Catalog,<sup>8,9</sup> the “info” URI registry,<sup>10</sup> and an SRW Registry,<sup>11</sup> among others. It also appears that ideas from the ERRoL system are being used by the Los Alamos National Laboratories, Research Library in their digital asset repository project.<sup>12,13</sup>

## ***Name Authority Control Files and other Registries***

One type of application where the OAI-PMH is finding use is with name authority control files and other similar registries. Not only is the OAI-PMH being used to disseminate or export records from these control files or registries, but using ERROl and similar systems, the OAI-PMH is being used as a significant part of the underlying architecture for some of these systems. The registry of OAI data providers at the UIUC and the ERROl system have already been described as an example of the types of synergies that can emerge between registry-type systems and the OAI-PMH.

One example of a name authority control file that utilizes OAI-based technology is OCLC's LC Name Authority Service<sup>14</sup> and Linked Authority File (LAF).<sup>15</sup> The LC Name Authority Service is a web service which allows interactive and automated queries of the Library of Congress' authority file. Interestingly, the identifiers of the returned name authority records are ERROls, such as <http://errol.oclc.org/laf/nr95-9068>, and as previously described, an ERROl is "cool URL" that refers to an item in an OAI repository. In this case, the ERROl refers to a name authority record in OCLC's LAF OAI repository, namely:

[http://alcme.oclc.org/laf/servlet/OAIHandler?  
verb=GetRecord&metadataPrefix=marcxml&identifier=nr95-9068](http://alcme.oclc.org/laf/servlet/OAIHandler?verb=GetRecord&metadataPrefix=marcxml&identifier=nr95-9068)

Currently, the LAF repository is not harvestable in that it will not respond to **ListIdentifier** or **ListRecords** verbs; it can only be used to retrieve records with the **GetRecord** verb for a given, known identifier. Systems such as the Networked Digital Library of Theses and Dissertations (NDLTD)<sup>16</sup> and various Institutional Repositories such as DSpace and Fedora are also beginning to explore how they can discover and link to name authority records, and OAI-PMH repositories of such records along with systems like ERROl provide one compelling part of the solution.

In collaboration with the Library of Congress and Die Deutsche Bibliothek, OCLC is also working on a union authority file called the Virtual International Authority File (VIAF).<sup>17 18</sup> One plan for this authority file is to utilize a shared federation of OAI-PMH servers to maintain the shared authority records and make them accessible to users. National authority records would be available via national OAI-PMH data providers, with a central service provider harvesting and merging records from the separate national authority files. Some national name authority records are already available via the OAI-PMH for harvesting, for example the Hong Kong University of Science and Technology Library has a Name Access Control Repository<sup>19</sup> which is harvestable from the base URL: <http://lbxml.ust.hk/nac/oai.pl>.

In addition to name authority records, the OAI-PMH is also being used for similar controlled authority files, for example experimental OAI repositories of ISBNs and ISSNs have been developed. There is an OAI repository of Internet MIME Types. Researchers are also exploring OAI repositories for various controlled vocabularies such as MeSH and LCSH. [Maybe find some references to these types of repositories.] Indeed the OAI-PMH seems to be suited to providing many of the bits and pieces required for a complete digital library infrastructure.

### ***Traditional Web Crawlers and the OAI-PMH***

One of the original goals of the OAI-PMH was to make resources that were previously hidden or not widely accessible easier to discover and more widely available. Many of the repositories which have embraced the OAI-PMH were originally part of the “hidden or deep web” or that part of the web which was not accessible to traditional web crawlers, such as Google, AskJeeves, Inktomi, and others. These resources were usually “hidden” because they were behind firewalls, required registration in order to use, were only accessible through web forms, or were in some format not widely accessible to web crawlers, such as XML. The OAI-PMH goes a long

way in making these resources more accessible. However, most of these resources are still not discoverable via the traditional web search services for various reasons, with the primary reason being that most of these services have not yet embraced the OAI-PMH. This is beginning to change with some of the major search services starting to investigate the protocol.<sup>20 21</sup> However, until the OAI PMH is fully supported by the search world there are services like DP9 which is described below.

Another interesting development for web crawlers is *mod\_oai*, an Apache software module that will expose content accessible from Apache Web servers via the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). This project is interesting in that in addition to making it easier for data providers to create OAI-compliant repositories, it also aims to make the process of web crawling itself more efficient. Both DP9 and *mod\_oai* are discussed in more detail below.

## **DP9**

DP9<sup>22 23</sup> was developed jointly by the Old Dominion University Digital Library group and the Los Alamos National Laboratory Research Library. There is a freely available, open source implementation available from <http://dp9.sourceforge.net/>. The current DP9 entry page for web crawlers is located at <http://arc.cs.odu.edu:8080/dp9/index.jsp>.

Basically, DP9 acts as a gateway between OAI data providers and web crawlers. It does this by maintaining a list of OAI data providers for which it is acting as a gateway. This list is exposed as a top-level HTML web page which is the starting page for any web crawler. From the entry page, DP9 dynamically creates a series of HTML web pages which if followed by a traditional web crawler will allow the crawler to index a complete OAI repository. The intermediate HTML pages are generated from cached data collected from ListIdentifier responses

from the OAI repositories. However, the HTML pages with the metadata are queried in real time from the actual OAI data provider and transformed from XML into HTML by XSLT. This means that the quality of service for any specific OAI data provider is dependent on the availability and quality of the original OAI data provider and not on the DP9 service itself. Similar to ERRoL, DP9 defines a persistent URL for an OAI record:

“http://” + <domain> + “/dp9/getrecord/” + <metadataPrefix> + “/” + <oai-identifier>

The DP9 service will resolve this URL by converting it into an OAI **GetRecord** request for the given identifier in the given metadata format. The resulting XML OAI response is converted by XSLT into an HTML page before being returned to the user or web crawler. The basic architecture is shown in Figure 1.

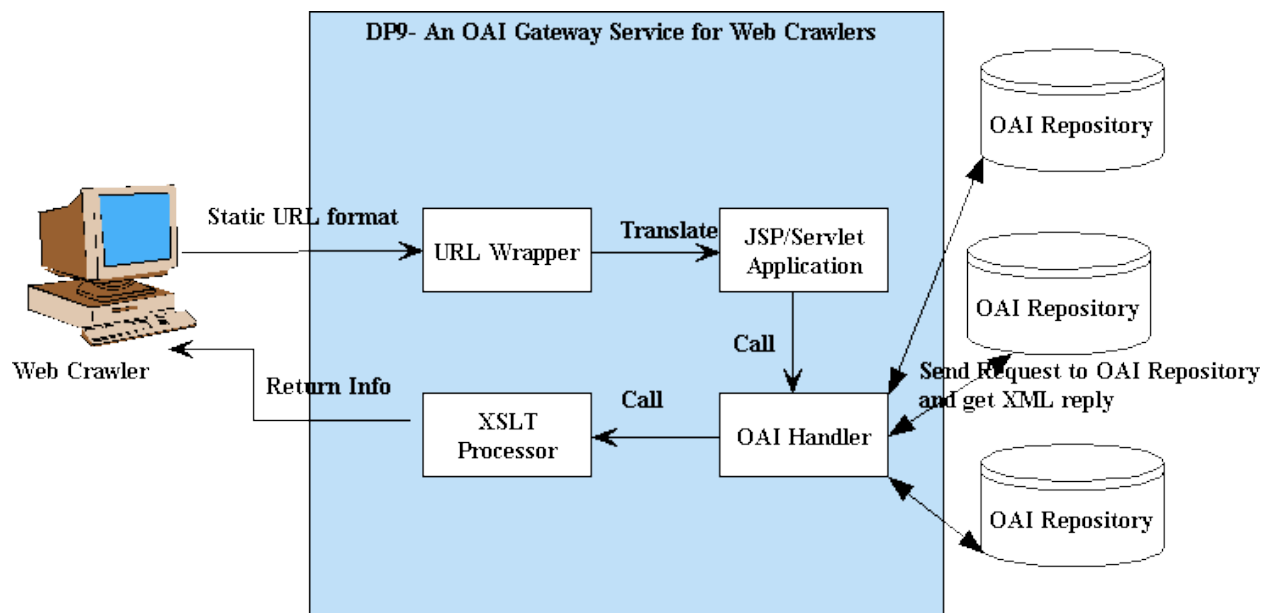


Figure 1: Architecture of the DP9 OAI Gateway Service for Web Crawlers (from <http://arc.cs.odu.edu:8080/dp9/about.jsp#how>) TBD: May need to get permission or draw our own picture.

Since its inception, DP9 or related gateways have been adopted by other OAI aggregators in order to facilitate the indexing of their resources by the traditional web search services. One example is the Open Language Archives Community (OLAC).<sup>24</sup> The ERRoL-based OAI Viewer<sup>25</sup>

at OCLC can also serve as a web crawler gateway. Essentially, any OAI aggregator which allows its harvested records to be browsed over the web as HTML pages, such as the Celestial<sup>26</sup> system from University of Southampton, can act as a web crawler gateway for OAI records. DP9 was the first system to implement this concept.

## **mod\_oai**

The mod\_oai<sup>27</sup> project was announced in early 2004<sup>28</sup> to create an Apache HTTP Server<sup>29</sup> software module that will expose content accessible from Apache web servers via the OAI-PMH. The project is funded by The Mellon Foundation and intends to release the mod\_oai software under the GNU Public License (GPL) Open Source license. As of the writing of this chapter there was one available demonstration server<sup>30</sup> and no publicly available implementations. However, the project has generated a large amount of interest not only from the Open Archives and Digital Library communities, but also from the web community in general<sup>31 32 33</sup> with news of the project spreading quickly through digital library blogging community.

There are several reasons that the project has garnered such attention. Perhaps the most obvious reason is the ubiquity of the Apache web server. Recent surveys place its usage at 68% of all web servers.<sup>34</sup> If mod\_oai proves successful, it has the potential to greatly increase the number of OAI-PMH data providers, making every Apache web server into an OAI data provider with little or no extra cost to the web server administrators. This will also make it much easier for potential repositories to become OAI-PMH compliant with any digital resources which are available via the web server also being available via the OAI-PMH.

Possibly the most compelling use case for mod\_oai is its potential for improving the efficiency of regular web crawling. Currently web crawlers must periodically revisit each resource hosted on a web server to determine whether the resource has changed since the last web crawl. If



the resource has changed the crawler must reindex and reparse the resource looking for links to additional new resources that it should also index. Even using common optimizations such as the HTTP HEAD request, this can be very inefficient, especially for web sites that change slowly or infrequently. However, `mod_oai` would allow a web crawler to issue OAI-PMH requests to the web server to much more efficiently determine which pages have changed or been added since the last web crawl. For example, using the **ListIdentifiers** request and the optional **from** parameter, a web crawler could with a minimal number of transactions receive a complete list of all resources at the web site which have been added or modified since a given date. Because `mod_oai` uses MIME types to create OAI-PMH sets, a web crawler could also use the optional **set** parameter to further limit the list to only certain MIME types of interest, such as PDF (`application/pdf`) or Microsoft Word (`application/msword`) documents. Since `mod_oai` uses the URL of the resource as the OAI identifier, a web crawler, after listing the modified resources, can use regular HTTP GET requests to retrieve the resources for indexing. However, `mod_oai` can also improve the actual retrieval of the resources themselves which is the third significant use case.

The third significant use case for `mod_oai` is in harvesting complete digital resources not just metadata about those resources. A web crawler could issue either **GetRecord** or **ListRecords** requests with a **metadataPrefix** of **oai\_didl** to retrieve the complete digital resources as MPEG-21 Digital Item Declaration Language (DIDL) documents.<sup>35</sup> DIDL “specifies a uniform and flexible abstraction and interoperable schema for declaring the structure and makeup of Digital Items.” Because of the ability to specify date ranges (the **from** and **until** parameters) and MIME types (the **set** parameter), the **ListRecords** request offers an especially efficient opportunity for web crawlers to harvest the complete collection of resources from a web site. A separate section of this chapter is devoted to using the OAI-PMH to harvest complete digital resources.

Because mod\_oai is not suitable for dynamically generated resources and because the metadata for the resources that are available from the web server are currently limited to what would normally appear in the HTTP header, such as date stamps, MIME types, file sizes, and identifiers, mod\_oai may not be a good replacement for the large metadata repositories which have been the traditional purview of the OAI-PMH. However, because of the ubiquity of the Apache web server and the potential of mod\_oai to revolutionize traditional web crawling, mod\_oai is a project worth monitoring.

### ***Using OAI-PMH to Harvest Complete Digital Resources***

Early in the development of the OAI-PMH it was recognized that it would be desirable to use the protocol to allow access to complete digital resources and not just metadata about those resources. No explicit provisions were added to the protocol for this, but neither were restrictions on full-text access made part of the protocol. Shortly after the protocol was released people began experimenting with options for using the protocol for harvesting complete resources. Essentially three approaches have emerged.

The first approach relies on the metadata to provide pointers to the location from which the digital resource can be obtained. This is probably the most fragile of the approaches in that it relies on the quality of the metadata which can vary widely across repositories or even within a single repository. The most basic application of this approach is parsing the oai\_dc metadata looking for URLs in the <dc:identifier>, <dc:relation>, or other fields. The first problem with this is that many records will have multiple URLs in different fields, and the question becomes which identifier points to the “official” digital resource being described by the metadata. The different identifiers might even point to different versions of the same resource, violating the one-to-one principle.<sup>36</sup> The second problem is that often the URLs do not point to the digital resource at all, but to some

surrogate object or splash page for the object, making it nearly impossible to determine how to obtain the resource. Various refinements to this approach have been proposed<sup>37 38 39</sup> mostly dealing with standardizing on different metadata profiles or metadata best practices. These include proposals to always include the URL to the “official” digital resource as the first <dc:identifier> in the oai\_dc record. If the URL in the oai\_dc record points to a splash page instead of to the actual digital resource, other suggestions are that the splash page should contain special link elements in the HTML head that point to the digital resource, such as:

```
<link rel="alternate" class="fulltext" type="[MIME type]" href="[URL]" title="Full Text ([mime type])" />
```

Probably the best refinement to this approach is to develop specific metadata formats beyond simple oai\_dc that explicitly identify the URL to the digital resource, such as proposals for Qualified Dublin Core that suggest using the <dcterms:hasFormat> element to point to different versions of the resource, such as:

```
<dcterms:hasFormat  
xsi:type="dcterms:URI">http://eprints.bath.ac.uk/12345.html</dcterms:hasFormat>
```

The distinguishing characteristic of all of the above options is that they rely on a community of users to agree on and follow whatever scheme is decided upon for identifying the digital resource.

The second approach adds extensions to the OAI-PMH itself to be used to request the retrieval of digital resources. For example, OA-X<sup>40</sup> is an extension of the OAI-PMH protocol being developed for the i-Tor<sup>41</sup> project. One of the extensions defined by OA-X is a new **verb=GetObject** that would return the metadata and the digital resource in one response, such as:

```
<OAI-PMH ...>  
<responseDate>2004-06-10T12:10:19Z</responseDate>  
<request metadataPrefix='oai_dc' verb='GetObject'>...some url...</request>  
<GetObject>  
  <record>  
    <header>...</header>  
    <metadata>
```

```

<article xmlns:dc=...>
  <dc:creator>D.J.Tiesto</dc:creator>
  <dc:title>The effects of filtered bass on the mendula oblangata</dc:title>
  <dc:identifier>art-01-06763</dc:identifier>
</article>
</metadata>
<object contentType='application/pdf' name='art-01-06763.pdf'>
FJIDSIJBBVJDBSLVJDLFJVDJBVKBEOPGYGFERUIHFSNDKJBCJNBWEIHIF
BJKFNBWEIHBIFJWEBFIJEWBIFNSDBWOUEBOJQJBOJBCBOWEIJHFSJDH
...
BFOJOJBCOIJENBWOJNFOKEWNOJEWNCIJBWEBFOWBEHOFJWEBJKNOE
ENOJFIOEWJBFOIJEWNFOJEWNFJWNOJFNOWEJFNOJWEBJWEWBIFJEBS
</object>
</record>
</GetObject>
</OAI-PMH>

```

Here the <object> element contains the bitstream for the digital object encoded using base64Binary<sup>42</sup> encoding along with some attributes such as the content MIME type and the name of the object. A major limitation of this approach is that it can only handle simple resources composed of a single bitstream.

The third approach actually packages the digital resources and returns them as part of the regular OAI **GetRecord** or **ListRecords** responses. This is the approach being taken by the Los Alamos National Laboratory Digital Library.<sup>43</sup> This approach relies on special XML Schemas that allow complete digital resources to be described and packaged in a single XML record. These XML schemas are exposed by the OAI-PMH simply as another of the available metadata formats returned by the **ListMetadataFormats** verb. These schemas must not only support simple objects composed of a single data stream, such as a PDF document, but they must also support complex or compound digital objects, consisting of multiple separate data streams, such as an HTML page and all of its associated images, style sheets, and so forth. Other examples of compound digital objects include a series of scanned page images of a book. The scanned book is the digital object, but it is composed of possibly hundreds of separate data streams, one for each scanned page image. Not

only must the individual data streams be maintained, but also metadata about each stream, especially the relationships between the data streams, such as which page image follows which other page image, in the example of the scanned book. A number of architectures and schemas have emerged that address these issues.<sup>44 45 46</sup> Several have been considered for use with the OAI-PMH, including METS and MPEG-21 DIDL.

The Metadata Encoding and Transmission Standard (METS)<sup>47</sup> is funded by the Digital Library Federation (DLF) and maintained by the Network Development and MARC Standards Office of the Library of Congress. It is described as “a standard for encoding descriptive, administrative, and structural metadata regarding objects within a digital library.” METS documents consist of seven major sections: 1) METS Header, 2) Descriptive Metadata, 3) Administrative Metadata, 4) File Section, 5) Structural Map, 6) Structural Links, and 7) Behavior. The first three sections contain various sorts of metadata about the objects. The metadata can be included by reference via pointers or it can be included inline in various formats. The File Section contains either pointers to or the base64 encoding of the data streams. The Structural Map and Structural Links sections define the relations between different data streams and the links between them. The Behavior section associates executable behaviors with the METS contents. Although, METS has been adopted by a number of digital libraries<sup>48</sup> and institutional repository systems, such as DSpace,<sup>49</sup> Fedora,<sup>50</sup> and OCLC’s Digital Archive,<sup>51</sup> its use to disseminate digital resources via the OAI-PMH is limited to a few experimental systems.

A competing standard is the MPEG-21 Digital Item Declaration Language (DIDL) which is a part of the MPEG-21 Multimedia Framework, developed by the Moving Picture Experts Group,<sup>52</sup> a working group of ISO/IEC. Although the MPEG group is primarily interested in digital audio and video, much of the MPEG-21 standard is broadly applicable to any complex digital objects,

such as electronic books and journals or scientific datasets. DIDL is similar to METS. However, not only is DIDL part of the already successful MPEG effort, but DIDL has a potential advantage of being part of a larger framework which attempts to address other issues relevant to digital libraries, such as object identification, intellectual property, and rights expression. Plus, DIDL is the schema being used by the LANL Digital Library which has noted that the MPEG-21 Framework already has a lot in common with several other developing digital library frameworks.

53 54

In addition to the standard METS and DIDL XML schemas there are also other alternatives which may emerge. Some repositories have chosen to define their own XML Schemas for disseminating their full-text; one example is BioMed Central.<sup>55</sup> Plus there are other approaches which could be used as well, such as the IMS Content Packaging XML Binding,<sup>56</sup> the Sharable Content Object Reference Model (SCORM),<sup>57</sup> and the XML packaging approach developed by CCSDS Panel 2.<sup>58</sup>

### ***OAI-PMH for Individuals and Small Groups***

Even though the OAI-PMH was designed specifically as a simple to implement, low cost protocol (especially for data providers) there are still significant barriers to entry for individuals or small organizations with limited resources. Examples of these barriers include technical expertise, financial resources, organizational inertia, or inability to control their own web server. The following two systems have been developed to address the needs of these small data providers.

### **Kepler**

The first of these is the Kepler<sup>59 60 61</sup> system developed by the Old Dominion University Digital Library Research Group. Kepler was designed for small publishers or even individuals

who have a small number of ten to a hundred resources that they would like to publish or archive. One potential user group is individual researchers who want to self publish or self archive their own working papers or e-prints. An implementation of the complete Kepler system is available as a free, open source download from SourceForge.<sup>62</sup>

Kepler is based on peer-to-peer network models. It is composed of a combination OAI Compliant Repository and Publishing Tool, called an *Archivelet*, a Registration Service, and a Service Provider, as shown in Figure 2. The registration service registers new archivelets and keeps track of their current status, namely their address and whether they are currently active. Whenever an archivelet is started or stopped it will notify the registration service. A service provider will query the registration service in order to locate archivelets which are active. Once located, a service provider can either harvest metadata from an active archivelet or retrieve complete digital resources. An archivelet is a downloadable application consisting of an OAI-compliant data provider and a publication tool. The archivelet provides a simple interface allowing the user to register the archivelet and specify metadata and upload files into the archive which is simply stored on the local machine's disk. Once the files are described and uploaded into the archivelet, they are harvestable directly from the user's workstation by service providers.

Kepler currently utilizes the Arc OAI Service Provider<sup>63</sup> also developed by the Old Dominion University Digital Library Research Group. However, it is possible to run an archivelet for use by any service providers without registering with a registration service. However, in order for the archivelet to be utilized by other service providers there must be some way to notify those providers of the existence and state of the archivelet.

An extended version of the Kepler Framework was developed about a year after the original version. The basic architecture is the same. However, the extended framework adds

features that allow archivelets to be both data providers and service providers. In other words, archivelets may harvest data from other archivelets. This allows archivelets to be “buddy nodes” or “super nodes” with each other allowing them to replicate each others’ data, see Figure 3. This makes for higher availability of data, as well as other advanced features. In addition, the extended framework adds features to allow archivelets to either notify service providers of new data or actually push the data to the service providers, unlike the typical OAI-PMH model where all data is pulled from data providers.

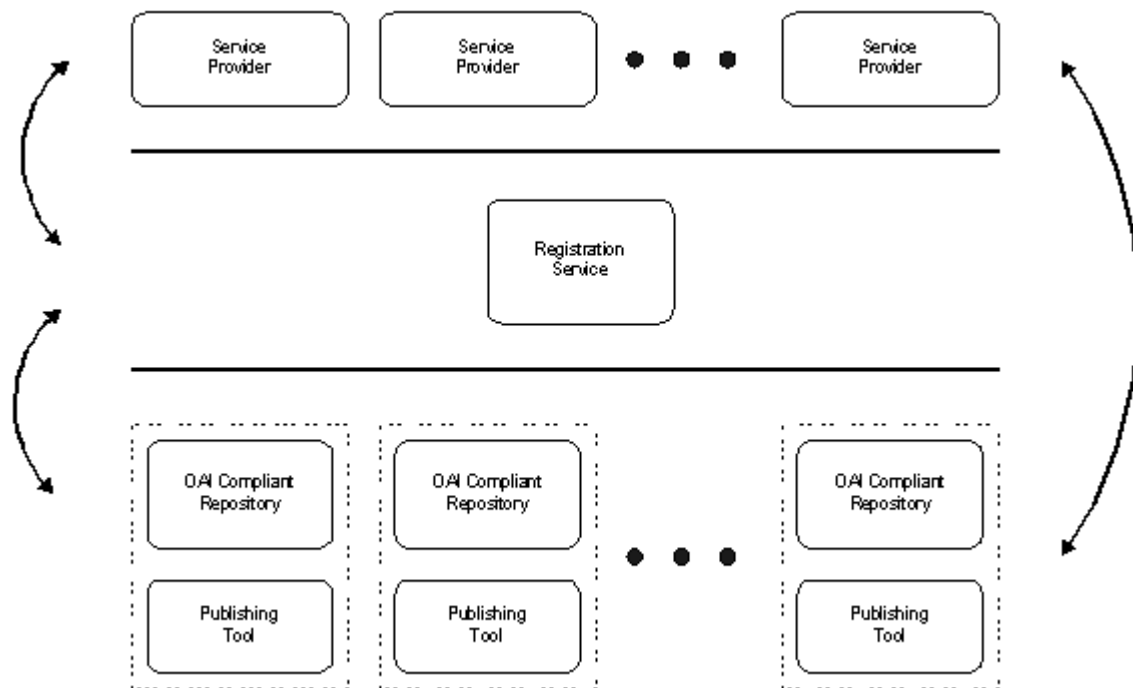


Figure 2: Kepler Framework (from <http://www.dlib.org/dlib/april01/maly/04maly.html>) TBD: May need to get permission or draw our own picture.



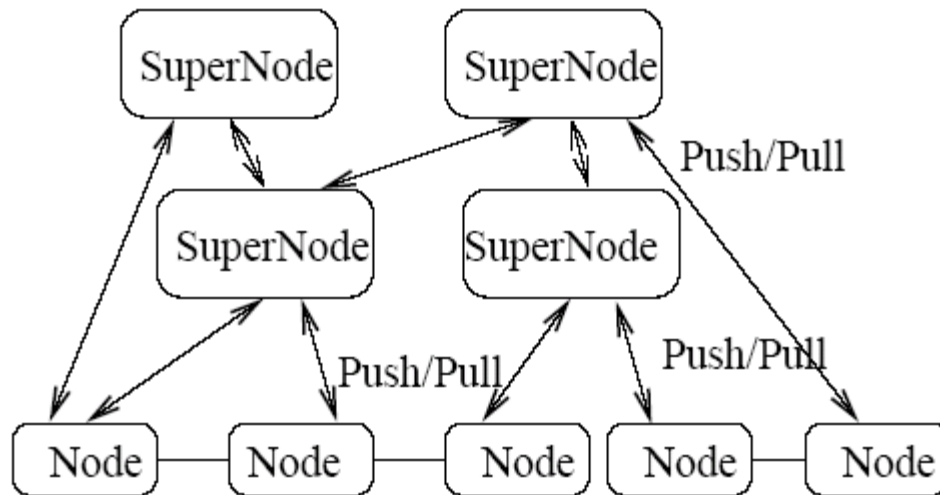


Figure 3: Extended Kepler Architecture (from [http://www.cs.odu.edu/~liu\\_x/paper/kepler/kepler.pdf](http://www.cs.odu.edu/~liu_x/paper/kepler/kepler.pdf))  
 TBD: May need to get permission or draw our own picture.

## OAI Static Repository and Static Repository Gateway

Another extension to the OAI-PMH that was specifically designed to lower the barriers to entry is the OAI Static Repository and OAI Static Repository Gateway.<sup>64 65</sup> It was designed for repositories 1) having metadata collections ranging in size between 1 and 5000 records which do not change often, and 2) which can make static XML content available through a network-accessible web server, and 3) which need a technically simpler implementation strategy than the typical OAI-PMH data provider. The OAI Static Repository specification is part of the Implementation Guidelines for the Open Archives Initiative Protocol for Metadata Harvesting.<sup>66</sup> The specification consists of two parts, the Static Repository itself, and the Static Repository Gateway.

The static repository itself is nothing more than a single XML file containing all of the metadata, identifiers, and date stamps for all items in the repository, plus data needed to respond to the Identify and ListMetadataFormats requests, such as (for clarity, namespaces are not shown):

```

<Repository>
  <Identify>
    ...
  </Identify>
  <ListMetadataFormats>
    ...
  </ListMetadataFormats>
  <ListRecords metadataPrefix="oai_dc">
    ...
  </ListRecords>
  <ListRecords metadataPrefix="other">
    ...
  </ListRecords>
  ...
</Repository>

```

A static repository may support multiple metadata formats for each item. However, static repositories do not support OAI sets or deleted records. The static repository XML file must be accessible from a web server via an HTTP URL, such as <http://some.host.edu/path/file.xml> with a MIME type of text/xml, and the XML file must have a character encoding of UTF-8. This XML file must also conform to the static repository XML Schema at <http://www.openarchives.org/OAI/2.0/static-repository.xsd>. Generally, the static XML file is created and maintained by the owner of the repository. This file can be created and maintained in various ways from using a simple text editor or a special purpose XML editor to being programmatically derived from a database or transformed from some other pre-existing XML files.

The second part of the specification is the static repository gateway. A static gateway provides intermediation for one or more static repositories, as shown in Figure 4.

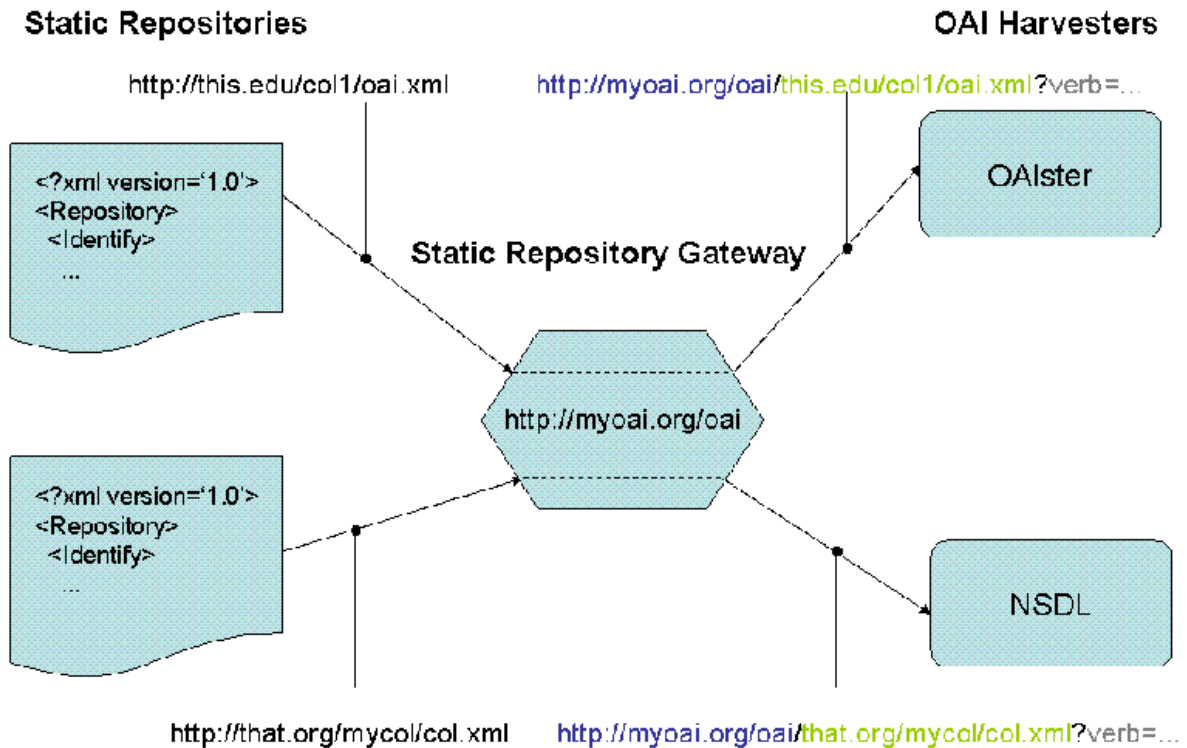


Figure 4: Static Repositories, a Static Repository Gateway and an OAI-PMH Harvester TBD: Drawn myself, need to make B&W version

Intermediation with a gateway is initiated by a static repository owner or administrator by issuing an **initiate** command to the gateway, such as:

`http://myoai.org/oai?initiate=http://this.edu/col1/oai.xml.`

Note that the command is issued to the base URL of the static gateway, and the value of the **initiate** parameter is the base URL of the static XML file. There are also a couple ways to terminate intermediation with a gateway. One way is to simply delete the static XML file, the other way is to change the base URL in the static XML file, and then issue a **terminate** command, such as:

`http://myoai.org/oai?terminate=http://this.edu/col1/oai.xml.`

Terminating would be done to take down a repository, but it should also be done before initiating a static repository with a new gateway, since in order to prevent duplicate records from being propagated, a given static repository should only be intermediated by a single gateway.

As part of the initiate process, the gateway will load the static XML file from the given URL. The gateway will ensure that the base URL in the Identify section of the static XML file is the concatenation of the static gateway's base URL and the URL of the static XML file, such as:

```
<oai:baseURL>http://myoai.org/oai/this.edu/col1/oai.xml</oai:baseURL>
```

This ensures that a single static XML is only intermediated by single gateway at any one time. In addition, different gateways may also have other policies about how many or what type of static repositories they will intermediate, for example, some gateways may only intermediate static repositories which are part of a given community of repositories. The best example of this type of gateway is the Open Language Archives Community (OLAC) Static Repository Gateway.<sup>67 68</sup>

In addition to the OLAC gateway, other examples of static gateways are the experimental ones operated by the Los Alamos National Laboratory (LANL) Research Library<sup>69</sup> and the University of Illinois at Urbana-Champaign (UIUC) Grainger Engineering Library Information Center.<sup>70</sup> There are also two free, open source static gateway implementations currently available. One was developed by the LANL Research Library and is available from <http://srepod.sourceforge.net/>, and the other was developed by the UIUC Grainger Engineering Library Information Center and is available from <http://uilib-oai.sourceforge.net/>.

OAI static repositories are an easy entry into the OAI world. All that is needed is a valid XML file that sits on an accessible HTTP web server. Just register with a gateway and the static repository becomes an instant OAI data provider. Unfortunately, there are not yet many general purpose gateways, but hopefully this will improve as more small repositories discover the static protocol.

## ***Future of the OAI Protocol***

The OAI protocol is still fairly young with the 1.0 version officially released early in 2001, followed six months later by the 1.1 version, and finally by version 2.0 about a year after that in the middle of 2002. There have also been several minor clarifications and bug fixes to the 2.0 version of the protocol since it was released.

It is likely that there will be some additional enhancements to the protocol in coming years. The following two sections are speculative, but they will briefly describe some possible directions that enhancements or changes to the protocol could take. The first section will discuss the Simple Object Access Protocol's (SOAP) use with the OAI-PMH. The second section will discuss various miscellaneous proposed extensions to the OAI-PMH that have appeared in the literature. While neither SOAP nor any of the enhancements discussed in the second subsection appear to be headed toward "official" standardization as part of the OAI-PMH, they are frequent topics of discussion in OAI-related mailing list and research papers, and therefore are worth monitoring.

### **Simple Object Access Protocol (SOAP)**

The Simple Object Access Protocol (SOAP) version 1.2 <sup>71</sup> is a recommended standard of the World Wide Web Consortium. Paraphrasing from the SOAP specification, the Simple Object Access Protocol is an XML-based, "lightweight protocol intended for exchanging structured information in a decentralized, distributed environment." This very much describes the OAI protocol itself, so it would seem that SOAP would be a natural fit for implementing the OAI-PMH, and SOAP was discussed by the OAI-PMH authors, especially during the development of the 2.0 version of the OAI-PMH.

However, the primary reason that SOAP was not originally used for the OAI protocol was that SOAP and OAI were developed somewhat contemporaneously. The original SOAP

specifications were published in early 2000, first as an “Internet-Draft” from the Internet Engineering Task Force (IETF), and later as a “Note” from the World Wide Web Consortium<sup>72</sup>, neither of which carry the weight of an officially sanctioned standard. SOAP did not become an officially sanctioned standard (recommendation) of the W3C until June, 2003. The authors were reluctant to base the OAI-PMH on an *informal* standard. In addition, even by the time that the OAI-PMH was being developed in early 2001, there was not a lot of support for SOAP. There were not many tools for software developers, and it was not supported across a range of programming languages and platforms; whereas, the standard HTTP CGI-style of web programming (also referred to as REST<sup>73 74</sup>) upon which the OAI-PMH is based was very widely supported and understood. Even though SOAP was not used for the OAI-PMH version 2.0, it clearly had an impact on the design, and several of the original OAI-PMH authors considered the protocol to be “SOAP-ready,”<sup>75</sup> and were already considering using SOAP for any future revisions to the protocol.<sup>76</sup> In addition, there currently exist at least a couple prototype implementation of a SOAP-based OAI-PMH.<sup>77 78</sup>

Given the above, it is likely that any future revisions to the OAI protocol would consider SOAP as an alternative to the current REST-based architecture. However, even if SOAP is adopted for the OAI-PMH it is likely that it would be added onto the OAI-PMH as an alternative to the existing REST approach as opposed to replacing it. Supporting dual architectures of SOAP and REST is not unprecedented and is exactly the approach being followed by another of the important, emerging digital library standards, Search and Retrieve Web/URL Service (SRW/SRU)<sup>79</sup> which supports both a REST approach (SRU) and a SOAP approach (SRW).

Among the advantages offered by SOAP is the integration of the OAI-PMH into the wider world of Web Services which appears to be slowly emerging with the advent of SOAP 1.2. This

includes services such as search and retrieval (SRW), name authority control, among others. Also with the advent of SOAP 1.2, the state of development tools that support SOAP has markedly improved, as well as the supported environments and platforms. Most programming languages and platforms now have at least some level of support for SOAP, making it easier to develop applications based on the SOAP standards.

However, there as of yet is no clear winner between the REST and SOAP approaches, so it is likely that even if a SOAP version of the OAI-PMH is standardized it will have to work side-by-side with the exiting REST based implementations.

## **Extensions to the OAI-PMH**

In addition to SOAP, many different extensions to the basic OAI-PMH have been proposed or implemented on an experimental basis since the protocol was first introduced. The following sections will highlight some of these proposals. Some of these have already been discussed in previous sections of this chapter, but are also briefly included here for completeness. Many of the following examples have been taken from the Open Digital Library (ODL)<sup>80</sup> project and the i-TOR OA-X<sup>40</sup> project which are both exploring ways of extending the OAI-PMH to better support the diverse requirements of complete digital library systems.

## **Searching**

Supporting search capabilities seems like a fairly natural extension to the OAI-PMH, and many people unfamiliar with the protocol seem to mistakenly believe that search is part of the protocol. However, several proposals to extend the protocol to support search have met a mixed reception within the OAI community. This is probably due to strong bias of the OAI-PMH

developers for keeping the protocol simple, and keeping it optimized for its originally intended purpose. However, this has not prevented a number of novel approaches for supporting search.

A common proposal for extending the protocol for search is to use the optional **set** parameter of either the **ListIdentifiers** or **ListRecords** verbs, for example:

`?verb=ListIdentifiers&metadataPrefix=oai_dc&set=some_type_of_query`

where *some\_type\_of\_query* represents the query to be performed. This approach has the advantage of requiring the least changes to the protocol; no new verbs or parameters are required to be added to the protocol. Essentially, the query dynamically generates an OAI set which is returned as a regular OAI response. However, this does violate the semantics of the OAI protocol in two areas. One is the **ListSets** verb which must return a list of all sets supported by the repository. However, if sets can be created dynamically by embedding a query in the **set** parameter this becomes impossible. The other problem is with the record header where in order to be compliant with the 2.0 version of the protocol each set to which a record belongs must be listed. Once again, this is not possible if sets can be dynamically created. Nonetheless, this approach is being used by some systems, notably the Open Digital Library (ODL)<sup>80</sup> system proposed by Hussein Suleman and Edward Fox.<sup>81 82</sup> The ODL system even has a suggested query syntax for use in the set parameter, `qlang/query/start/stop`, where `qlang` identifies the actual query language to be used, `query` is the query string itself whose syntax depends on the `qlang` part, and `start` and `stop` are the indexes of the first and last records to retrieve for the query.

There have also been proposals to add new verbs or parameters to the protocol. One example<sup>83</sup> proposes a **matching** parameter whose value would be a search query, such as:

`?verb=ListIdentifiers&metadataPrefix=oai_dc&matching=some_type_of_query`



These proposals have the disadvantage of requiring changes to the core OAI-PMH without any real advantages over the alternate idea of using the **set** parameter; although, they do preserve the original set semantics of the OAI-PMH.

Although, this is not technically an extension to the OAI-PMH, a promising development in the area of search is the SRU protocols.<sup>79</sup> SRU is the REST-based version of the SRW protocol which stands for Search Retrieve Web service, and is essentially the web-friendly replacement of the venerable Z39.50 protocol. Digital library systems are beginning to emerge that take advantage of both the OAI and the SRU protocols. Combining the use of these two protocols has the advantage that now each protocol is being used for what it was optimally designed for: the OAI-PMH for metadata harvesting and aggregation and SRU for search and retrieval. A number of these sorts of hybrid systems have been developed. Jeffery Young at OCLC has developed a web service that can act as a gateway between SRW/SRU systems and OAI harvesters, essentially turning any SRW/SRU service into an OAI-PMH data provider. Similarly, there is the ZMARCO<sup>84</sup> project and the Z39.50 OAI Gateway Profile<sup>85</sup> from the University of Illinois at Urbana-Champaign which provides a similar gateway service between Z39.50 servers and the OAI-PMH. In addition, many OAI-based aggregations are utilizing the SRW/SRU protocol as a means to allow search and retrieval across the aggregations. With systems like this, it is easy to image these two protocols being merged to provide the best of both worlds.

## **Retrieving Resources**

The idea of retrieving complete digital resources using the OAI-PMH was covered in the Using OAI-PMH to Harvest Complete Digital Resources section of this chapter. The only proposal that really extended the protocol itself was for the addition of a new **GetObject** verb which is part of the OA-X proposal.<sup>86</sup>

## Adding Resources

The compliment of retrieving resources from a repository is adding resources, and for that purpose some implementers have proposed a new **PutObject** verb be added to the OAI-PMH. The OA-X proposal has also suggested this addition to the protocol.<sup>87</sup> The parameters required for the **PutObject** verb include **metadataPrefix**, **setSpec**, **metadata**, **identifier**, and **object**. The **PutObject** request would need to use the HTTP POST action because the **metadata** and **object** parameter values are the complete metadata and bit stream for the digital object and would be too long to include part of the URL query string.

Similarly, the ODL project has proposed a **PutRecord** verb as the compliment to the existing **GetRecord** verb. **PutRecord** would allow new metadata records to be submitted to a repository. This verb would support these parameters: **identifier**, **sets**, **metadataPrefix**, **metadata**, and **status**.

## Authentication

The Taiwan National Digital Archives Program (NDAP)<sup>88</sup> has proposed<sup>89</sup> some extensions to the OAI-PMH to support authenticated and secure metadata harvesting. By introducing a harvester registration service which keeps a log of all prospective harvesters of a given data provider their system can provide authentication and encryption services to data repositories to ensure that only authorized harvesters are allowed to harvest them. In addition, the harvester registry acts as a cryptographic key registry, assigning key pairs to every registered harvester to facilitate secure encrypted communication between harvester and data provider. To support secure harvesting using the registry, they have added a single optional parameter to all OAI requests, **hname**, which stands for harvester name. When a data provider encounters the **hname** parameter it can use harvester name to look up the encryption parameters for that harvester and use those

parameters to encrypt all OAI responses to that harvester. Details on their proposal can be found in their research paper.<sup>89</sup>

### **Subscription and Notification**

The OAI-PMH is essentially a pull-style protocol in that data harvesters are required to periodically pull data from repositories. This is a simple model and requires the least amount of effort for data providers, but it has some drawbacks mostly dealing with inefficiencies. There is no easy way for harvesters to know how frequently a given repository is being updated, so they will often either attempt to pull data too frequently resulting in unnecessary processing, both for the harvester and the data provider, or not frequently enough resulting in staleness of the harvested aggregation. A common solution to this problem is to introduce the concept of subscription and notification. In this model, a harvester would “subscribe” to a certain OAI repository, and it then becomes the responsibility of the data provider to “notify” subscribing harvesters whenever records have been added or modified. Once notified, the harvester can proceed to harvest the repository normally. Another more advanced option is that instead of just notify subscribing harvesters, the data provider actually “pushes” the new or modified data directly to the harvesters. Given the design philosophy of the OAI-PMH to keep the data provider half of the protocol as simple as possible, this has the problem of adding significantly to the complexity of data providers. Nonetheless, a number of systems have been proposed or developed that follow this model.

One such system is the Taiwan NDAP88<sup>89</sup> system. One advantage of this system is that the harvester registry which was described in the previous Authentication section can be used to mediate the subscriptions and notifications between harvesters and data providers, requiring fewer modifications to the data providers in order to support this model.

There are also other models for solving the problems associated with a pull-style protocol. In addition to the subscribe and notify model, a paper by Xiaoming Liu, et al.<sup>90</sup> also described methods that harvesters can use to estimate a data providers update frequency over a series of several harvests. This paper also describes a new optional **Identify** description container called syndication (taken from the RDF Site Syndication (RSS) specification<sup>91</sup>) that could be used by data providers to communicate to harvesters what their approximate update frequency is, such as:

```
<oai:description>
  <syndication xmlns="http://purl.org/rss/1.0/modules/syndication/">
    <updatePeriod>weekly</updatePeriod>
    <updateFrequency>1</updateFrequency>
    <updateBase>2004-01-01</updateBase>
  </syndication>
</oai:description>
```

The estimation or syndication approaches have the advantage of not requiring any changes to the OAI-PMH, but they still are not as efficient as the subscribe and notify models.

## **Conclusion**

A common measure applied to any new or emerging technology is how rapidly that technology is put to new and novel uses that were never envisioned by its developers. Based on the above sampling of some of these uses, the OAI-PMH has a bright future indeed. Not only has the uptake of the protocol for its intended purpose been fairly rapid and widespread, but it is finding wide use in applications and environments which are quite surprising.

- <sup>1</sup> Jeffery A. Young. Extensible Repository Resource Locators (ERRoLs) for OAI Identifiers, OCLC, <http://errol.oclc.org/>
- <sup>2</sup> <http://www.w3.org/Provider/Style/URI.html>
- <sup>3</sup> Andy Powell, Jeff Young, and Thom Hickey, "The PURL-based Object Identifier (POI)"
- <sup>4</sup> Herbert Van de Sompel, Jeffery A. Young, and Thomas B. Hickey. "Using the OAI-PMH ... Differently," D-Lib Magazine, 9, 7/8, July/August, 2003, Corporation for National Research Initiatives. <http://www.dlib.org/dlib/july03/young/07young.html>
- <sup>5</sup> <http://gita.grainger.uiuc.edu/registry/>
- <sup>6</sup> <http://www.openarchives.org/OAI/2.0/guidelines-oai-identifier.htm>
- <sup>7</sup> <http://www.oclc.org/research/projects/oairesolver/#CoordinatingContentInOAIRepositories>
- <sup>8</sup> Carol Jean Godby, Jeffrey A. Young, and Eric Childress. "A Repository of Metadata Crosswalks," D-Lib Magazine, December 2004, Volume 10 Number 12 <http://www.dlib.org/dlib/december04/godby/12godby.html>
- <sup>9</sup> <http://errol.oclc.org/schemaTrans.oclc.org.html>
- <sup>10</sup> <http://info-uri.info/registry/>
- <sup>11</sup> <http://errol.oclc.org/srwRegistry.oclc.org.html>
- <sup>12</sup> Jerez, H.N.; Liu, X.; Hochstenbach, P. & Sompel, H.V.d. "The multi-faceted use of the OAI-PMH in the lanl repository," JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries ACM Press, 2004, 11-20 <http://80-doi.acm.org.proxy2.library.uiuc.edu/10.1145/996350.996355>
- <sup>13</sup> Herbert Van de Sompel, Michael L. Nelson, Carl Lagoze, and Simeon Warner. "Resource Harvesting within the OAI-PMH Framework," D-Lib Magazine, 10, 12, December, 2004, Corporation for National Research Initiatives. <http://www.dlib.org/dlib/december04/vandesompel/12vandesompel.html>
- <sup>14</sup> <http://www.oclc.org/research/researchworks/authority/default.htm>
- <sup>15</sup> <http://errol.oclc.org/laf?verb=Identify>
- <sup>16</sup> <http://www.ndltd.org/>
- <sup>17</sup> <http://www.oclc.org/research/projects/viaf/default.htm>
- <sup>18</sup> <http://www.ifla.org/IV/ifla67/papers/094-152ae.pdf>
- <sup>19</sup> <http://library.ust.hk/info/nac/>
- <sup>20</sup> University of Michigan News Service, "U-M expands access to hidden electronic resources with OAIster," March 10, 2004 <http://www.umich.edu/news/index.html?Releases/2004/Mar04/r031004>
- <sup>21</sup> Chronicle of Higher Education, "Libraries Aim to Widen Google's Eyes," May 21, 2004 <http://chronicle.com/weekly/v50/i37/37a00101.htm>
- <sup>22</sup> Xiaoming Liu, Kurt Maly, Mohammad Zubair, and Michael Nelson. "DP9: An OAI Gateway Service for Web Crawlers," in Proceedings of the Second ACM/IEEE Joint Conference on Digital Libraries, 2002. <http://doi.acm.org/10.1145/544220.544284>
- <sup>23</sup> Xiaoming Liu, "In Brief: DP9 Service Provider for Web Crawlers," D-Lib Magazine, 7, 12, December, 2001, Corporation for National Research Initiatives. <http://www.dlib.org/dlib/december01/12inbrief.html#LIU>

- <sup>24</sup> OLAC web crawler gateway, <http://www.language-archives.org/dp9/>
- <sup>25</sup> <http://errol.oclc.org/>
- <sup>26</sup> <http://celestial.eprints.org/>
- <sup>27</sup> <http://www.modoi.org/>
- <sup>28</sup> “OAI News: mod\_oai Project Aims at Optimizing Web Crawling,” [http://www.openarchives.org/news/news2.html#mod\\_oai](http://www.openarchives.org/news/news2.html#mod_oai)
- <sup>29</sup> <http://httpd.apache.org/>
- <sup>30</sup> <http://whiskey.cs.odu.edu/>
- <sup>31</sup> Herbert Van de Sompel, Carl Lagoze, Simeon Warner, Michael L. Nelson, “An Update from the OAI: Presentation Session & Discussion Session,” presented at The CNI Fall 2004 Task Force Meeting, December 6-7, 2004, Portland, Oregon. [http://www.cni.org/tfms/2004b.fall/abstracts/handouts/CNI\\_sompel\\_OAI.pdf](http://www.cni.org/tfms/2004b.fall/abstracts/handouts/CNI_sompel_OAI.pdf)  
[http://www.cni.org/tfms/2004b.fall/abstracts/presentations/CNI\\_sompel\\_OAI.ppt](http://www.cni.org/tfms/2004b.fall/abstracts/presentations/CNI_sompel_OAI.ppt)
- <sup>32</sup> Michael L. Nelson, Herbert Van de Sompel, Xiaoming Liu, Aravind Elango, “mod\_oai: Metadata Harvesting for Everyone,” presented at DLF 2004 Fall Forum, Baltimore MD, October 25-27, 2004.  
<http://www.cs.odu.edu/~mln/pubs/dlf-2004/modoi.ppt>
- <sup>33</sup> Phillip Long, Infectious Adoption, Syllabus Magazine, July 3, 2004 <http://www.campus-technology.com/article.asp?id=9680>
- <sup>34</sup> Netcraft Web Server Survey [http://news.netcraft.com/archives/web\\_server\\_survey.html](http://news.netcraft.com/archives/web_server_survey.html)
- <sup>35</sup> <http://xml.coverpages.org/mpeg21-didl.html>
- <sup>36</sup> <http://dublincore.org/documents/usageguide/#whatis>
- <sup>37</sup> <http://www.rdn.ac.uk/projects/eprints-uk/docs/encoding-fulltext-links/>
- <sup>38</sup> Automatically gathering the full-text of eprints <http://www.openarchives.org/pipermail/oai-implementers/2004-March/thread.html#11184>
- <sup>39</sup> <http://oai-best.comm.nsdsl.org/cgi-bin/wiki.pl?IdentifyingTheResource>
- <sup>40</sup> [http://www.i-tor.org/oa\\_x/](http://www.i-tor.org/oa_x/)
- <sup>41</sup> <http://www.i-tor.org/en/>
- <sup>42</sup> <http://www.w3.org/TR/xmlschema-2/#base64Binary>
- <sup>43</sup> Herbert Van de Sompel, Michael L. Nelson, Carl Lagoze, and Simeon Warner, “Resource Harvesting within the OAI-PMH Framework,” D-Lib Magazine, December 2004, Volume 10 Number 12  
<http://www.dlib.org/dlib/december04/vandesompel/12vandesompel.html>
- <sup>44</sup> Michael L. Nelson, Brad Argue, Miles Efron, Sheila Denn, and Maria Cristina Pattuelli. “A Survey of Complex Object Technologies for Digital Libraries,” December, 2001, NASA/TM-2001-211426.  
<http://techreports.larc.nasa.gov/ltrs/PDF/2001/tm/NASA-2001-tm211426.pdf>
- <sup>45</sup> <http://library.nyu.edu/diglib/StructuredContent/>
- <sup>46</sup> <http://www.ccsds.org/CCSDS/documents/650x0b1.pdf>

<sup>47</sup> <http://www.loc.gov/standards/mets/>

<sup>48</sup> <http://sunsite.berkeley.edu/mets/registry/>

<sup>49</sup> <http://dspace.org/index.html>

<sup>50</sup> <http://www.fedora.info/>

<sup>51</sup> <http://www.oclc.org/digitalarchive/default.htm>

<sup>52</sup> <http://www.chiariglione.org/mpeg/>

<sup>53</sup> Jeroen Bekaert, Patrick Hochstenbach, and Herbert Van de Sompel, "Using MPEG-21 DIDL to Represent Complex Digital Objects in the Los Alamos National Laboratory Digital Library," D-Lib Magazine, November 2003, Volume 9 Number 11 <http://www.dlib.org/dlib/november03/bekaert/11bekaert.html>

<sup>54</sup> Jeroen Bekaert, Lyudmila Balakireva, Patrick Hochstenbach and Herbert Van de Sompel, "Using MPEG-21 DIP and NISO OpenURL for the Dynamic Dissemination of Complex Digital Objects in the Los Alamos National Laboratory Digital Library," D-Lib Magazine, February 2004, Volume 10 Number 2 <http://www.dlib.org/dlib/february04/bekaert/02bekaert.html>

<sup>55</sup> <http://www.biomedcentral.com/xml/>

<sup>56</sup> <http://www.imsglobal.org/content/packaging/>

<sup>57</sup> <http://www.adlnet.org/index.cfm?fuseaction=scormabt>

<sup>58</sup> <http://www.ccsds.org/docu/dscgi/ds.py/View/Collection-241>

<sup>59</sup> <http://kepler.cs.odu.edu/>

<sup>60</sup> Kurt Maly, Mohammad Zubair, and Xiaoming Liu. "Kepler – An OAI Data/Service Provider for the Individual," D-Lib Magazine, 7, 4, April, 2001, Corporation for National Research Initiatives. <http://www.dlib.org/dlib/april01/maly/04maly.html>

<sup>61</sup> Xiaoming Liu, Kurt Maly, Mohammad Zubair. "Enhanced Kepler Framework for Self-Archiving," in Workshop on Distributed Computing Architectures for Digital Libraries. ICPP 2002. <http://citeseer.ist.psu.edu/liu02enhanced.html>

<sup>62</sup> [http://sourceforge.net/project/showfiles.php?group\\_id=107191&package\\_id=115659&release\\_id=233083](http://sourceforge.net/project/showfiles.php?group_id=107191&package_id=115659&release_id=233083)

<sup>63</sup> Xiaoming Liu, Kurt Maly, Mohammad Zubair. Michael L. Nelson. "Arc - An OAI Service Provider for Digital Library Federation," D-Lib Magazine, 7, 4, April, 2001, Corporation for National Research Initiatives. <http://www.dlib.org/dlib/april01/liu/04liu.html>

<sup>64</sup> Herbert Van de Sompel, Carl Lagoze, Michael Nelson, Simeon Warner, Patrick Hochstenbach, Henry Jerez "Specification for an OAI Static Repository and an OAI Static Repository Gateway," <http://www.openarchives.org/OAI/2.0/guidelines-static-repository.htm>

<sup>65</sup> Patrick Hochstenbach and Henry Jerez and Herbert Van de Sompel "The OAI-PMH static repository and static repository gateway," in JCDL '03: Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries. IEEE Computer Society. 2003

<sup>66</sup> <http://www.openarchives.org/OAI/2.0/guidelines.htm>

<sup>67</sup> <http://www.language-archives.org/sr>

- <sup>68</sup> Steven Bird and Gary Simons “Building an Open Language Archives Community on the DC Foundation” in: Hillmann and Westbrook (editors) *Metadata in Practice: A Work in Progress*, ALA Editions, 2004. <http://www.language-archives.org/documents/mip.pdf>
- <sup>69</sup> <http://libtest.lanl.gov/registry.html>
- <sup>70</sup> <http://imlsdcc.grainger.uiuc.edu/gateway/oai.asp>
- <sup>71</sup> Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, and Henrik Frystyk Nielsen (eds), “SOAP Version 1.2 Part 1: Messaging Framework,” World Wide Web Consortium, W3C Recommendation 24 June 2003 <http://www.w3.org/TR/soap12> .
- <sup>72</sup> D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F.Nielson, S. Thatte, and D. Winer, “Simple Object Access Protocol (SOAP)1.1,” World Wide Web Consortium, W3C Note May 08 2000. <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/> .
- <sup>73</sup> REST is an acronym which stands for *RE*presentational *S*tate *T*ransfer. REST was first coined in Roy Fielding’s PhD dissertation which was an attempt to describe the Web’s architectural style. REST proponents promote the idea that HTTP, URIs, and XML are adequate for most Web applications or services. A common attribute of REST architectures is using URIs to convey the complete semantics of a transaction, very much in the way that the OAI-PMH uses a baseURL and query string parameters.
- <sup>74</sup> Roy Thomas Fielding, “Architectural Styles and the Design of Network-based Software Architectures,” PhD. Thesis, University of California, Irvine, 2000. <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- <sup>75</sup> Michael L. Nelson, Herbert Van de Sompel, and Simeon Warner, “Advanced Overview of Version 2.0 of the Open Archives Initiative Protocol for Metadata Harvesting,” ACM/IEEE Joint Conference on Digital Libraries, Houston, Texas, May 27 2003. <http://www.cs.odu.edu/~mln/jcdl03/oai-2.0-adv.ppt>
- <sup>76</sup> Herbert Van de Sompel and Carl Lagoze. "Notes from the Interoperability Front: A Progress Report on the Open Archives Initiative," Research and Advances Technology for Digital Technology : 6th European Conference, ECDL 2002, Rome, Italy, September 16-18, 2002. Proceedings, M. Agosti, C. Thanos (Eds.): pp. 144 – 157 <http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2458&page=144>
- <sup>77</sup> Rufeng Liu, “SOAP-based Implementation of OAI,” Dr. Mohammad Zubair (MS Project Advisor), Computer Science Department, Old Dominion University, 2003. [http://www.cs.odu.edu/~rl/OAI\\_SOAP/gateway.html](http://www.cs.odu.edu/~rl/OAI_SOAP/gateway.html)
- <sup>78</sup> Congia, S., M. Gaylord, B. Merchant and H. Suleman. “Applying SOAP to OAI-PMH,” in Proceedings of 8th European Conference on Research and Advanced Technology for Digital Libraries (ECDL2004), Bath, UK, 12-17 September 2004. [http://aim.cs.uct.ac.za/tiki-download\\_file.php?fileId=110](http://aim.cs.uct.ac.za/tiki-download_file.php?fileId=110)
- <sup>79</sup> Library of Congress, “SRW/SRU Version 1.1 13th February 2004,” <http://www.loc.gov/z3950/agency/zing/srw/>
- <sup>80</sup> <http://oai.dlib.vt.edu/odl/>
- <sup>81</sup> Hussein Suleman and Edward Fox, “Designing Protocols in Support of Digital Library Componentization” [http://www.husseinsspace.com/publications/ecdl\\_2002\\_paper\\_odl.pdf](http://www.husseinsspace.com/publications/ecdl_2002_paper_odl.pdf)
- <sup>82</sup> Hussein Suleman and Edward Fox, “Beyond Harvesting: Digital Library Components as OAI Extensions” [http://www.husseinsspace.com/publications/cstr\\_2002\\_odl\\_1.pdf](http://www.husseinsspace.com/publications/cstr_2002_odl_1.pdf)
- <sup>83</sup> <http://documents.cern.ch/ettdh/doc/public/OAIRSF.html>
- <sup>84</sup> <http://zmarco.sourceforge.net/>
- <sup>85</sup> <http://frasier.library.uiuc.edu/research.htm>
- <sup>86</sup> [http://www.i-tor.org/oa\\_x/retrieving\\_objects/](http://www.i-tor.org/oa_x/retrieving_objects/)



<sup>87</sup> [http://www.i-tor.org/oa\\_x/uploading\\_resources/](http://www.i-tor.org/oa_x/uploading_resources/)

<sup>88</sup> [http://www.ndap.org.tw/index\\_en.php](http://www.ndap.org.tw/index_en.php)

<sup>89</sup> Chao-Chin Chou, Pei-Xian Kuo, Jan-Ming Ho, D.T. Lee “Union Catalog Using Extended OAI-PMH”  
<http://www.iis.sinica.edu.tw/~ccchou/publications/oai.pdf>

<sup>90</sup> Xiaoming Liu, Kurt Maly, Mohammad Zubair, and Michael L. Nelson , “Repository Synchronization in the OAI Framework,” [http://www.cs.odu.edu/~liu\\_x/paper/freshness/freshness.pdf](http://www.cs.odu.edu/~liu_x/paper/freshness/freshness.pdf)

<sup>91</sup> <http://web.resource.org/rss/1.0/modules/syndication/>